Sanjit Chatterjee
Palash Sarkar

Identity-Based Encryption

Springer

Chapter 1
Introduction

1.1 Background

Science, it has been argued by Kuhn in 1962 [126], advances through paradigm shifts. Concepts emerge that open-up new vistas of research, fundamentally changing the way we are used to looking at things. Between these paradigm shifts remain the periods of consolidation. Periods when human mind explores the newly found territory, shedding light on hitherto unknown dimensions. If radical changes are the hallmarks of paradigm shifts, the period within witnesses small but continuous developments, occasionally marked by its own milestones. It is in these periods that human faculty tries to grasp the full significance of the new concepts, consolidates its gains and thereby pushes the boundary of our collective knowledge further. The prospects, nevertheless, bring with it new problems too. Perhaps, by the way, making ground for the next paradigm shift.

Cryptology, as a branch of human knowledge, is no exception to this common story. Whenever civilisation reached a certain level of sophistication, the need for secret communication between two geographically distant parties has arisen. Politics, military and business are the three dominant areas of human activity where such communication becomes essential. With the gradual evolution of technology from ancient days to modern times, several innovative forms of encryption methodology have been developed, extensively used and then discarded after brilliant insights into new cryptanalytic methods. The cycle of development and analysis of cryptographic schemes have thus progressed to become a fascinating subject of human intellectual endeavor. There are excellent accounts of this historical development in [115, 164].

The basic problem of cryptography can be considered to be the issue of secure communication between two parties using a communication medium which is not under the exclusive control of these two parties. Such a medium is called a public channel, highlighting the fact that the information flowing across this channel is publicly accessible. An alternate view of cryptography is the task of building an implicit secure communication channel over an explicitly given insecure public

S. Chatterjee and P. Sarkar, *Identity-Based Encryption*, DOI 10.1007/978-1-4419-9383-0_1,

channel. Figure 1.1 provides an overview of the classical model of performing this task.

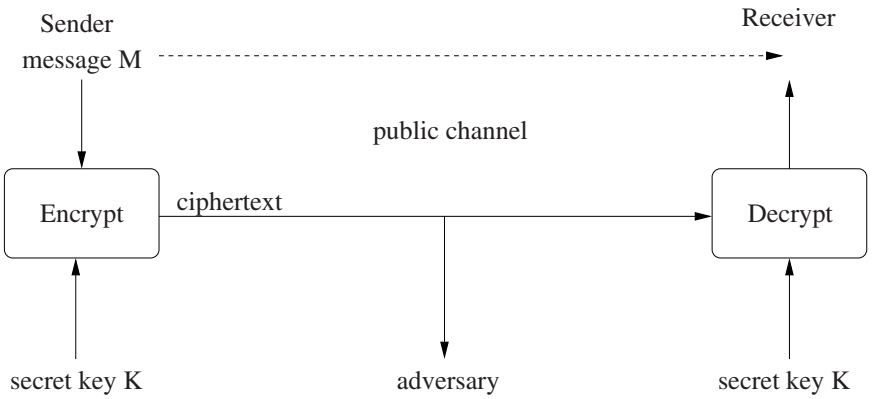

Fig. 1.1 An overview of symmetric key encryption.

Both the sender and the receiver share a common secret key K which is not known to the adversary. The ciphertext moving across the public channel is a function of the message and the secret key K. An adversary has access to the ciphertext, but, without knowledge of K should be unable to obtain the intended message M. The receiver, on the other hand, knows K and should be able to recover M from the ciphertext.

Since antiquity, human mind had accepted this model as the natural and perhaps the only model of secure communication. There had been no reason or motivation to look beyond this model. Things, however, changed with the advent of radio communication. This enabled large-scale communication. But, the full functionality of radio communication could not be realised if the confidentiality of the communication could not be assured. Thus, arose the problem of ensuring secure communication between any two of a number of parties.

Suppose there are 100 parties. Using the classical model, secure communication between any two parties requires a secret key per pair of parties. So, the total number of secret keys in the system is $\binom{100}{2}$ and each party has to maintain 99 secret keys. See Figure 1.2 for a pictorial representation of this scenario. To visualise the immensity of the problem, one may change 100 to any number n that would be practical in the real world.

Necessity is the mother of invention (attributed to the Greek philosopher Plato) and this is exemplified in the further development of cryptography. While previously, there had been no reason to consider anything but the classical model, with the advent of radio and the concomitant revolution in communication, arose the necessity of developing manageable methods of ensuring security of such communications. Solution to this problem was made possible through a paradigm shift in the discipline of cryptology.

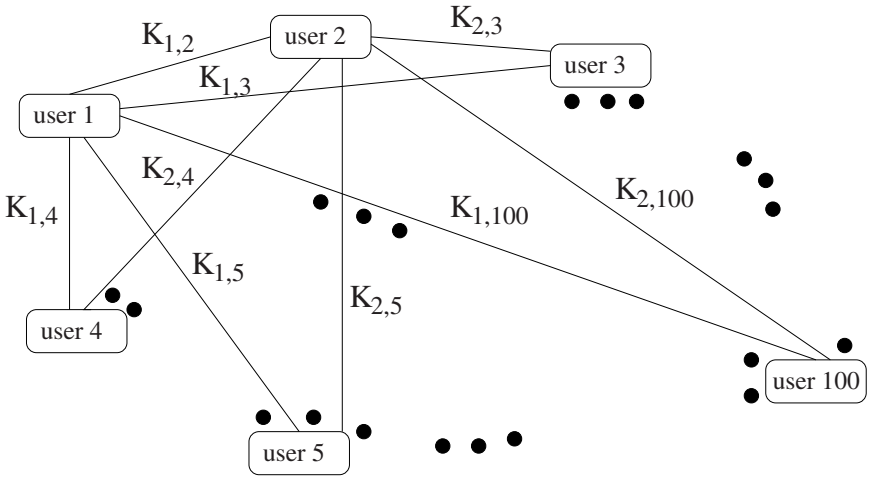

Fig. 1.2 Secure communication among 100 users using the classical model.

The basic idea of the new paradigm was simple. Instead of using the same key for encryption and decryption, one may consider two separate keys for each party – one key is used for encryption and the other key is used for decryption. The encryption key may be made public, so that any other party (say Alice) may send an encrypted message. The decryption key, on the other hand, should be kept secret, so that only the intended receiver (say Bob) can decrypt the ciphertext. See Figure 1.3 for an overview of this idea. This is called public key encryption (PKE).

Though it is surprisingly simple, the idea eluded researchers for a long time. This perhaps lends credence to Kuhn's theory of major scientific developments (alluded to in the opening paragraph) as proceeding through paradigm shifts. The human mind had grown accustomed to the belief (or paradigm) of using the *same* key for both encryption and decryption and hence, found it extremely difficult to conceive the shift where the encryption and decryption keys are different.

It was first published by Diffie and Hellman in their seminal paper [77] appropriately titled "New Directions in Cryptography". Somewhat interestingly, researchers working for the British government had also obtained the same idea but, their work remain classified for several decades. See [164] for an account of the two separate histories of the development of public key cryptography (PKC).

Though the concept of PKE was introduced by Diffie and Hellman, they were unable to provide a concrete instantiation of such a scheme. It was left as an open problem until it was solved by three other researchers (Rivest, Shamir and Adleman [148]) and was called the RSA public key encryption system. Diffie and Hellman had introduced and solved another related and equally important problem. They considered the possibility of two parties performing some private computations and exchanging some messages over a public channel to finally arrive at a shared secret key. This is called public key agreement. The Diffie-Hellman key agreement (DH-KA) is shown in Figure 1.4.

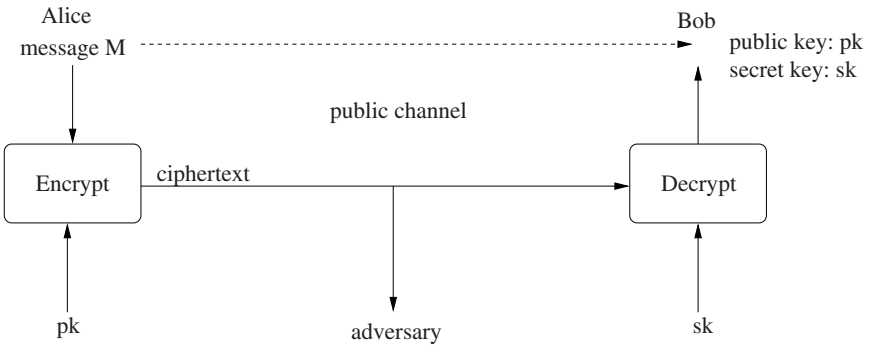

Fig. 1.3 An overview of public key encryption.

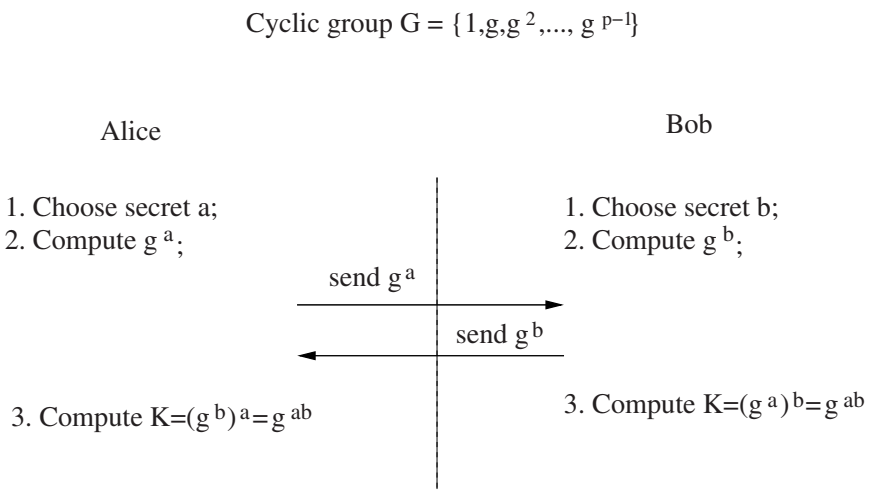

Fig. 1.4 Diffie-Hellman Public Key Agreement.

Note that the security of the Diffie-Hellman scheme relies on the facts that given g^a and g^b, no third party will be able to compute (i) one of the exponents a or b or (ii) the shared secret g^{ab}. The former is a classical problem in number theory called the discrete logarithm problem and the latter has become known as the Diffie-Hellman problem. So, the assurance that Alice and Bob get is conditioned on the fact that no one can solve these problems within a reasonable time. In fact, the security of all public key cryptographic schemes is based on the assumed hardness of some computational problem(s).

Later a public key encryption scheme was developed by ElGamal [79] which is very closely related to the Diffie-Hellman key agreement (DH-KA) protocol. This scheme is shown in Figure 1.5. This can be seen as a modification of the DH-KA. Bob does his part of the DH-KA protocol offline during the set-up phase and publishes (g, g^b) as his public key. Alice does her part of the DH-KA protocol online

to compute g^a and the common secret key g^{ab}. This secret key is used to mask the message. Bob can recover the message by computing g^{ab} from g^a and his secret key b as in the DH-KA protocol.

It is indicative of the subtlety of the problem that Diffie and Hellman having proposed the notion of PKE and having discovered their key agreement protocol could not get the PKE scheme later proposed by ElGamal. It is also perhaps an issue of (a smaller scale) paradigm shift. In the ElGamal PKE scheme, the ciphertext consists of two group elements, but, the message is a single group element. So, there is a ciphertext expansion. In all previous encryption schemes, including the RSA scheme, the ciphertext is as long as the message. Though it is now considered routine, at that point of time in history, it was perhaps difficult to conceive a PKE scheme where the ciphertext is longer than the message.

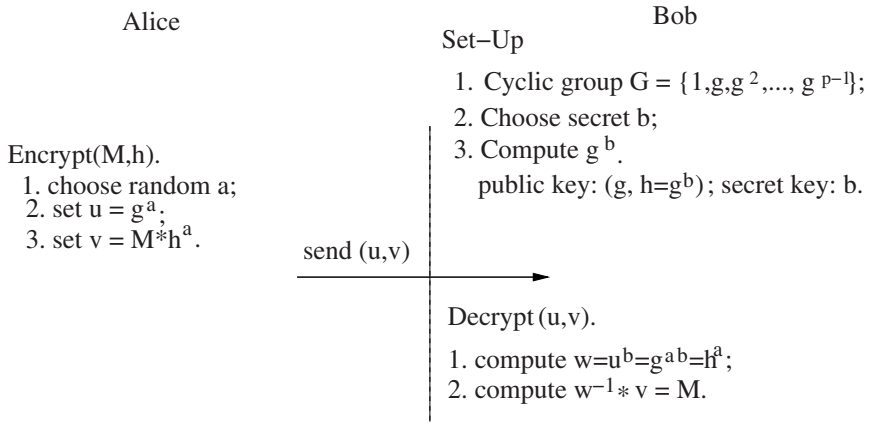

Fig. 1.5 ElGamal public key encryption scheme.

An important aspect of PKC is the notion of digital signatures. In this primitive, each user has a secret signing key and a public verification key. A message M is signed using the signing key to produce a signature σ. Anybody can verify the correctness of a message-signature pair using the public verification key. Concrete proposals of signature schemes were made using the RSA and the ElGamal PKE schemes. Figure 1.6 provides an overview of a digital signature scheme.

The advent of PKE is a major landmark in the evolution of cryptography. This, however, brought with it its own problems. Consider the situation shown in Figure 1.7. Here Eve is an active attacker, i.e., she can modify the information flowing across the public channel. So, she can intercept the messages that Alice and Bob exchange and replace them with messages of her own choosing. This allows her to establish secret keys separately with Alice and Bob without them realising this fact. In other words, Alice and Bob believe that they have established a shared key between themselves, whereas in actuality, they have established keys with Eve. So, if Alice sends a message to Bob using the key she believes that she has established

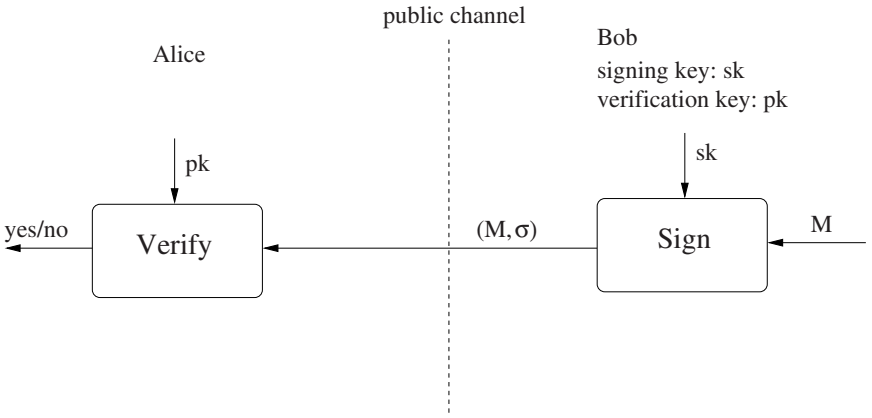

Fig. 1.6 Overview of a digital signature scheme.

with Bob, Eve can decrypt this message. The same is true if Bob sends an encrypted message to Alice. The crucial issue is that during key establishment, Alice and Bob do not authenticate themselves to each other.

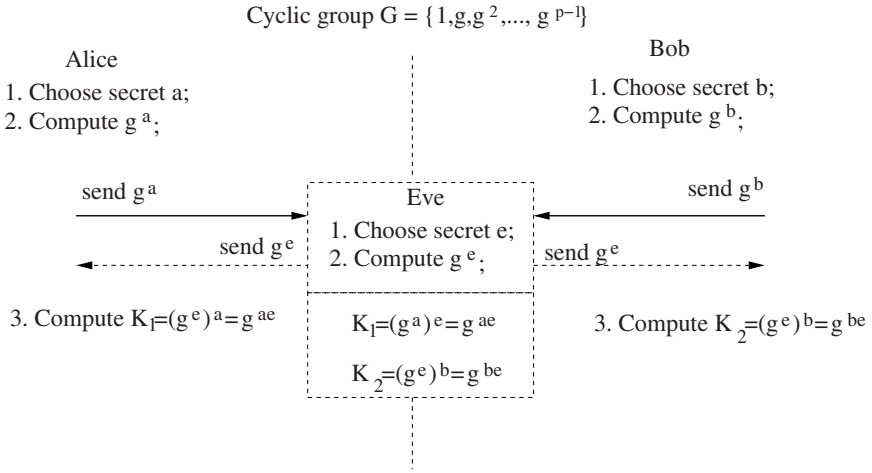

Fig. 1.7 (Wo)man in the middle attack.

PKE schemes are also subject to similar attacks. The issue is how to trust a public key? See Figure 1.8. When Bob wants to send a message to Alice, he will be using Alice's public key to encrypt the message. Suppose, Eve masquerades as Alice and puts forward a public key claiming it to belong to Alice. Eve, of course, knows the corresponding secret key. If Bob trusts this public key, then he will encrypt the message using this public key. Eve can decrypt the corresponding ciphertext thus, defeating the security of the system. To prevent this attack, Bob somehow needs to

be ensured that the public key that is claimed to belong to Alice indeed does belong to Alice.

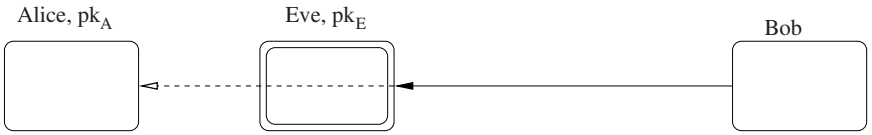

Fig. 1.8 How to trust a public key?

This is achieved using an authority that both Alice and Bob trust. This authority issues certificates for public keys and is called a certifying authority (CA). To obtain a certificate, Alice approaches the CA and submits her public key. The CA does the necessary (physical) verification and determines the identity of Alice. After such checking the CA uses a digital signature scheme to sign the public key of Alice. When Bob wishes to send a message to Alice, he first obtains Alice's public key and the certificate issued to Alice by the CA. Using the public verification key of the CA, Bob can verify the signature of CA on Alice's public key. If this verification succeeds, Bob trusts the public key to actually belong to Alice and uses it to encrypt a message intended for Alice. This is pictorially shown in Figure 1.9.

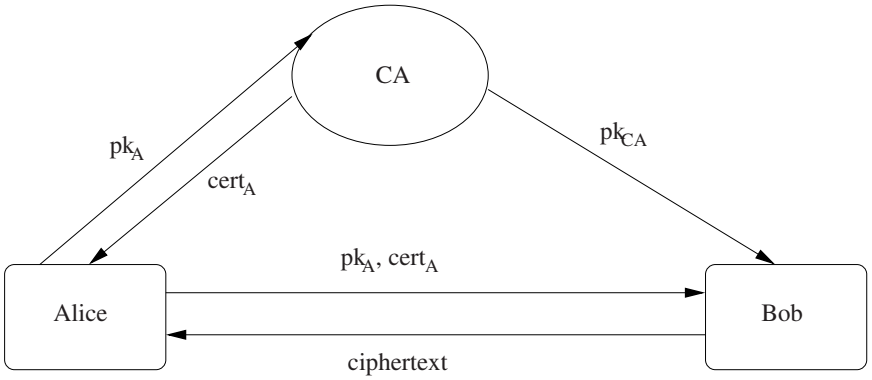

Fig. 1.9 Certifying authority and trust in public key.

Use of CA solves the problem of trust but, brings with it a host of other issues regarding certificates. The basic question is the validity of a certificate. A certificate might have been issued to Alice by the CA, but, this will have a time limit. Even otherwise, Alice may have compromised her secret key and has to obtain a new public key/secret key pair and so, a new certificate on the new public key. The issue of management of certificates is complex and cumbersome. There are no neat solutions known and the issue is the main stumbling block in the widespread deployment of PKE schemes.

1.2 Identity-Based Encryption

The problems associated with the practical deployment of PKE schemes motivated Shamir [155] to introduce the concept of identity-based encryption (IBE). IBE is a kind of public key encryption scheme where the public key of a user can be any arbitrary string – typically the e-mail address. An overview of an IBE scheme is given in Figure 1.10. When Bob wants to send a message to Alice; he encrypts it using the e-mail id of Alice as the public key. There is no need for Bob to go to the CA to verify the public key of Alice. This way an IBE can greatly simplify certificate management. To quote Shamir [155]:

> "*It makes the cryptographic aspects of the communication almost transparent to the user, and it can be used effectively even by laymen who know nothing about keys or protocols.*"

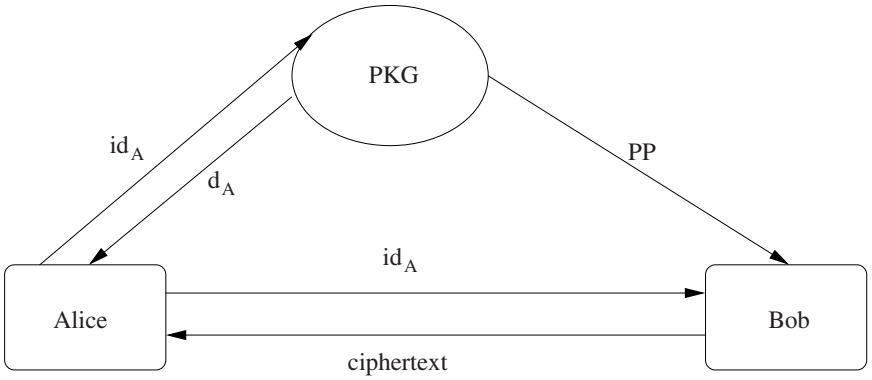

Fig. 1.10 An overview of an IBE scheme.

An IBE consists of four algorithms: Set-up which generates a set of public parameters together with a master key, Key-Gen which generates the private key of an entity, given her identity, Encrypt that encrypts a message given the identity and Decrypt that decrypts the message using the private key. Instead of a certifying authority, here we have a private key generator (PKG) who possesses the master secret key. In the above example, Bob authenticates himself to the PKG to obtain a private key corresponding to his identity. (This can be even after he receives the encrypted message from Alice.) Bob uses this private key to decrypt all the messages encrypted using his identity as the public key. Note that, Alice need not verify any certificate relating the public key to send an encrypted message to Bob. What she needs is the identity of Bob along with the public parameters of PKG.

Shamir posed a challenge to the crypto community to come out with a practical IBE scheme. A satisfactory solution of this problem eluded cryptographers till the turn of the millennium. The solution, when it finally arrived, came not from one, but three different quarters – Boneh-Franklin [39], Sakai-Ohgishi-Kasahara [151] and Cocks [70]. Among these, the former two based their cryptosystems on bilinear

pairing over elliptic curve groups while the last one was based on factoring and is less practical. Boneh and Franklin [39] formalised the notion of IBE, gave a precise definition of its security model and related the security of their construction with a natural analogue of the Diffie-Hellman problem in the bilinear setting, called the bilinear Diffie-Hellman (BDH) problem. This work caught the immediate attention of the crypto community world wide and turned out to be a major milestone within the paradigm of public key cryptography.

Boneh and Franklin use bilinear pairing over elliptic curve groups to construct their IBE. Initially bilinear maps such as Weil and Tate pairing were introduced in cryptology [133, 84] as weapons in the arsenal of the cryptanalyst – to reduce the discrete log problem in the elliptic curve group to the discrete log problem over finite fields. Joux [113] converted it into a tool of cryptography by proposing a one round tri-partite key agreement protocol using bilinear pairing. The works of Joux and Boneh-Franklin, in some sense, ignited an explosion in pairing based public key cryptography. Even a cursory glance at a compendium [17] of these works is enough to give a feel of the research activity that has been generated in the last one decade.

The concept of IBE was soon extended to hierarchical identity-based encryption (HIBE) [106, 94]. Requests for decryption keys are made to the PKG. A single PKG may become overloaded with such requests. The motivation for HIBE is to reduce the workload for the PKG. In a HIBE, instead of a single component, identities are considered to be vectors. Further, an entity having identity $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_j)$ and possessing the private key d_{id} can generate the private key of an entity whose identity is $\mathsf{id}' = (\mathsf{id}_1, \ldots, \mathsf{id}_j, \mathsf{id}_{j+1})$. This way HIBE reduces the workload of the PKG. The important thing to note is that the public parameters are that of the PKG, i.e., there are no lower level public parameters. Apart from being of some importance in its own right, a HIBE provides a flexible mechanism which can be used to construct other cryptographic primitives.

1.3 Plan of the Book

Technical discussion starts from the second chapter. Chapters 2 and 3 are of preliminary nature. Formal definitions of PKE and IBE schemes are given in Chapter 2. Extension to HIBE schemes are explained. The different variants of the security models for IBE and HIBE schemes are provided. Formal definition of IBE and an appropriate security model were proposed by Boneh and Franklin [39]. Later definitions and security models for related primitives were based on the foundation laid out in that work. An important aspect of the cryptographic schemes is the so-called security proofs or more accurately the security reductions. Chapter 2 discusses the overall structure of security proofs. This should help the reader in going through the proofs in the later chapters.

Most IBE schemes and certainly the practical ones are based on bilinear maps over elliptic curve groups. Discussing the relevant aspects of elliptic curves requires us to spend some space on finite field arithmetic. Chapter 3 provides the appropriate

material on finite fields, elliptic curves and pairing. This material is not intended to be encyclopedic. Rather it is provided so that the reader can get some feel about how IBE schemes can actually be implemented.

The first publicly known IBE scheme is due to Boneh and Franklin [39]. The BF-IBE scheme is presented in Chapter 4 and the security reduction is discussed in some details. The proof assumes certain functions to be random functions – the so-called random oracle assumption. All schemes discussed in Chapter 4 have this common feature. Gentry and Silverberg [94] had extended the BF-IBE scheme to a HIBE scheme. Several variants of the BF-IBE scheme has been given by other authors. Chapter 4 provides a good idea of such schemes.

Historically, the first attempts to remove the random oracle assumption from the security proofs required a weakening of the security model. This is called the selective-identity model. Several elegant (H)IBE schemes have been proposed and shown to be secure in the selective-identity model. Chapter 5 provides a description of two important (H)IBE schemes. The first one is due to Boneh and Boyen [32] and the second one is due to Boneh, Boyen and Goh [35].

Design of IBE schemes can be thought of as a two-step procedure. In the first step, a scheme is designed which can be proved to be secure against chosen-plaintext attacks (CPA-secure). The proof technique for such schemes incorporate mechanisms to handle key extraction procedures. The second step is to attain security against chosen-ciphertext attacks (CCA-secure). Several methods have been proposed to convert CPA-secure schemes to CCA-secure schemes. Chapter 6 provides a discussion of these methods.

The next major step is to obtain IBE schemes which can be proved to be secure in the model introduced by Boneh and Franklin without using the random oracle assumption. An early construction of such a scheme was given by Boneh and Boyen [33]. This scheme, however, was quite impractical. An important variant of this scheme was given by Waters [169]. This variant is one of the most important IBE schemes proposed till date. Its importance stems both from the theoretical novelty of the construction as well as from its practicality. The only disadvantage of this scheme is the rather large size of its public parameters.

Independent work by Chatterjee and Sarkar [60] and Naccache [137] showed how to reduce the size of the public parameters with an associated degradation in the security bound. Chapter 7 discusses in details Waters IBE scheme and its variants. A new proof of Waters' IBE scheme was given by Bellare and Ristenpart [23]. The proof of Waters' IBE scheme that we provide is based upon the approach in [23]. Chapter 7 also provides a HIBE scheme secure against adaptive-identity attacks and its modification to attain CCA-security. This discussion is based on [61, 154].

One of the most important issues about the schemes in Chapter 7 is that the hardness assumption is the decisional bilinear Diffie-Hellman (DBDH) problem. This is the decision version of the bilinear Diffie-Hellman (BDH) problem introduced by Boneh and Franklin in [39]. The BDH and DBDH problems are regarded as the basic hardness assumptions in pairing based cryptography.

Subsequent important work on pairing based (H)IBE schemes have been done by Gentry [91] and Waters [170] (Note that this is different from Waters [169] dis-

cussed in Chapter 7). Gentry's work is important as it is the simplest scheme which can be proved to be secure against adaptive-identity attacks without the use of random oracles. The drawback is that the hardness assumption that is used is much more complex compared to the DBDH assumption. Waters [170] recently introduced a new technique called dual system encryption for designing IBE schemes. This technique, though relatively new, has been used in subsequent works and appears to hold out promise of further applications. Chapter 8 discusses Gentry's [91] and Waters' [170] IBE schemes and some of their implications.

Though most IBE schemes proposed till date use bilinear pairings, there has been some work on non-pairing based IBE schemes. These schemes are discussed in Chapter 9. The first such scheme is due to Cocks [70] and is based on the hardness of the quadratic residuacity problem. One problem with Cocks' scheme is that the size of the ciphertext is quite large. Boneh, Gentry and Hamburg [40] proposed an IBE scheme which reduces the size of the ciphertext (at the cost of increasing the time for encryption).

Lattice based techniques have recently found applications to the design of IBE schemes. The possibility of lattice based IBE was first discovered by Gentry, Peikert and Vaikuntanathan [93]. Later work using lattices have paralleled more or less the development of pairing based IBE schemes. Chapter 9 provides a somewhat detailed description of the lattice based method for designing IBE schemes.

IBE by itself is an important cryptographic primitive. Its further importance arises from the fact that both IBE and HIBE have proved to be useful in designing other cryptographic primitives and new functionalities. These include signatures, key agreement, broadcast encryption and public key encryption with keyword search. Some idea of such applications are given in Chapter 10.

A nagging issue with IBE is that of key escrow. The PKG knows the decryption key for every identity and consequently can decrypt any ciphertext formed using its public parameters. For real-life applications this can be a drawback. (In some situations, though, this might be exactly what is required.) Chapter 11 discusses the several approaches for dealing with the issue of key escrow in IBE schemes.

Though the idea of IBE is only about a decade old, some commercial products have already appeared in the market. Additionally some standards have been proposed. Finally, Chapter 12 briefly mentions such products and standards.

Chapter 2
Definitions and Notations

In this chapter, we present definitions and notation. We start with the definition of public key encryption schemes and their security models. This forms the basis of the corresponding notions for identity-based encryption schemes. The definition of IBE schemes is given and extended to that of HIBE schemes. Security model for HIBE schemes is defined. This security model can be specialised to that of IBE schemes by fixing the number of levels to one.

There are several variants of the security model for (H)IBE schemes. These are carefully explained and the notion of anonymity is defined. A related issue is the use of random oracles in the security analysis. We mention this briefly and discuss its relevance.

2.1 Public Key Encryption

A public key encryption (PKE) scheme is specified by three probabilistic algorithms. The run-time of each of these algorithms is upper bounded by a polynomial in a quantity called the security parameter, denoted by κ. This is formally expressed by explicitly providing 1^κ as input to the algorithms and requiring the run-times of the algorithms to be upper bounded by a polynomial in the length of this input. While this is formally appropriate, it is more convenient to simply note that the run-times are polynomially bounded in κ and avoid explicitly mentioning this.

Set-Up. This algorithm takes as input a security parameter κ. It outputs descriptions of the message space, the ciphertext space, the key space and a key pair (pk, sk) from the key space. Here pk is a public key and sk is the corresponding secret key. The pair (pk, sk) is randomly sampled from the key space. (Though it is not a definitional requirement, (pk, sk) would typically be uniformly distributed over the key space.)

Encrypt. It takes as input a message M and a public key pk and outputs a ciphertext C.

S. Chatterjee and P. Sarkar, *Identity-Based Encryption*, DOI 10.1007/978-1-4419-9383-0_2,

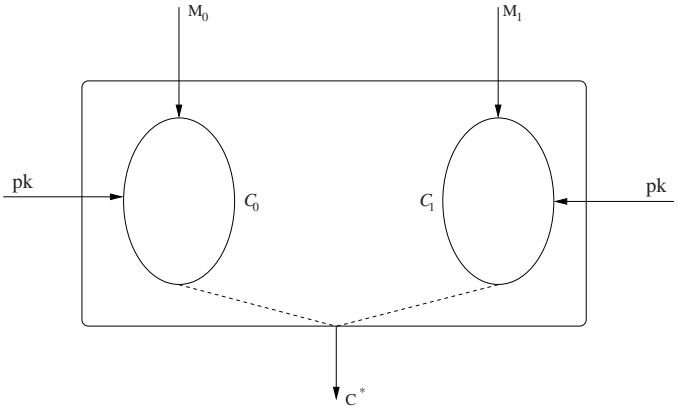

Fig. 2.1 $\mathscr{C}_0$ (resp. $\mathscr{C}_1$) corresponds to the set of possible ciphertexts that can arise when the encryption algorithm is applied to the message M_0 (resp. M_1). C^* is a uniform random choice from $\mathscr{C}_\gamma$, where γ is a uniform random bit.

Decrypt. It takes as input a ciphertext C and a private key sk and returns either a message M or the special symbol $\perp$. The symbol $\perp$ indicates that the ciphertext cannot be decrypted.

The encryption algorithm is a probabilistic algorithm and so there can be more than one ciphertext for a fixed message and a fixed public key. Equivalently, the encryption algorithm can be viewed as a sampling algorithm that given a message M and a public key pk samples from the set of possible ciphertexts which correspond to M and pk. Again the sampling will typically be done under the uniform distribution, though, it is not a definitional requirement.

A ciphertext can be said to be valid if it can be produced as an output of the encryption algorithm (on some pair of inputs M and pk) and invalid otherwise. The definition of the decryption algorithm does not require that the output has to be $\perp$ if the ciphertext is invalid; in this case, it may produce a random element of the message space as output.

For soundness, we require that if C is produced by Encrypt using pk, then the output of Decrypt on C using the corresponding secret key sk should give back M. Since the algorithms are probabilistic, the outputs are actually random variables over appropriate sets. In particular, the Set-Up algorithm can be seen to be sampling a pair of public and private keys from appropriate key spaces and the Encrypt algorithm samples from the set of possible ciphertexts which correspond to a message M and a public key pk. In principle, even though the Decrypt algorithm is allowed to be probabilistic, for most constructions, it is in fact a deterministic algorithm. We note that there are constructions, where the decryption algorithm is allowed to fail with an insignificant probability of error.

Next comes the question – how to define the security of a public key encryption scheme? A natural answer is – given a ciphertext no adversary should be able to learn any *meaningful* information about the corresponding plaintext. This intuitive

notion is formalised into what is called *semantic security* in a landmark paper by Goldwasser and Micali [99]. They also provided a technical definition of security called *indistinguishabilty* and showed that for a passive attacker these two notions are equivalent. This result has later been extended to the case of an active adversary in [98, 168]. The equivalence between the natural notion of security and the technical definition turns out to be very important. Because it is more convenient to work with the technical definition of indistingushability than the natural notion of semantic security.

This technical notion of indistinguishability of ciphertexts for a PKE scheme in the case of a passive adversary can be easily understood with the help of Figure 2.1. For $i = 0, 1$, let $\mathscr{C}_i$ be the set of ciphertexts which may arise from the message M_i under the public key pk. The encryption algorithm defines a distribution over $\mathscr{C}_i$. Suppose that a bit γ is chosen uniformly at random and a ciphertext C^* is sampled from $\mathscr{C}_\gamma$ according to the distribution defined by the encryption algorithm.

An adversary is allowed to specify the messages M_0 and M_1; the bit γ is not revealed to the adversary, but, the ciphertext C^* is given to the adversary. Now the adversary has to guess the value of γ. If the adversary is unable to do so (with probability significantly away from half), then, to the adversary, the ciphertexts arising from M_0 are indistinguishable from the ciphertexts arising from M_1. This basic idea is built into an appropriate security model as we describe below for an active adversary.

Indistinguishability against adaptive chosen ciphertext attack [21] is the strongest accepted notion of security for a public key encryption scheme. An encryption scheme secure against such an attack is said to be IND-CCA2 secure. We give an informal description of IND-CCA2 security in terms of the following game between a challenger and an adversary $\mathscr{A}$, which is a probabilistic algorithm whose runtime is bounded above by a polynomial in the security parameter. Later we provide a more detailed explanation of the security game for an IBE scheme. Figure 2.2 gives an overview of the security game for a PKE scheme.

1. Given the security parameter κ, the challenger runs the Set-Up algorithm to generate a public and private key pair (pk, sk). It gives $\mathscr{A}$ the public key pk.
2. Given the public key, $\mathscr{A}$ *adaptively* issues decryption queries, which the challenger must properly answer. By adaptively it is meant that the adversary's next query can depend on the answers to the previous queries.
3. At some point, $\mathscr{A}$ outputs two equal length messages M_0, M_1 and the challenger responds with an encryption C^* of M_γ, where γ is a random bit.
4. The adversary continues with adaptive decryption queries but not on C^*.
5. Finally, $\mathscr{A}$ outputs its guess γ' of γ and wins if $\gamma' = \gamma$.

The advantage of $\mathscr{A}$ against the encryption scheme is

$$\mathsf{Adv}_{\mathscr{A}} = \left| \Pr[\gamma = \gamma'] - \frac{1}{2} \right|.$$

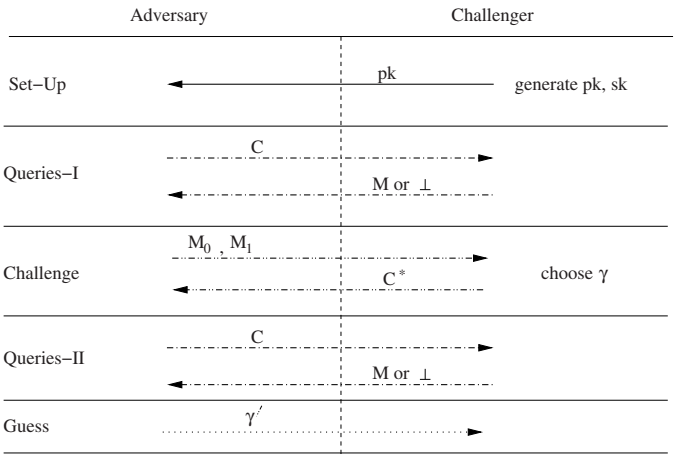

Fig. 2.2 A diagrammatic depiction of the five phases of the security model for a public key encryption scheme.

An encryption scheme is said to be (t,q,ε)-IND-CCA2 secure, if for all adversaries $\mathscr{A}$ running in time t and making at most q decryption queries, $\mathsf{Adv}_{\mathscr{A}} \leq \varepsilon$.

In case of a passive adversary, a weaker notion of security, called indistinguishability against chosen plaintext attack (in short IND-CPA security) of a public key encryption scheme is available in the literature [99, 21]. In the IND-CPA security game, the adversary is not allowed to place any decryption query. In other words, this is the scenario depicted in Figure 2.1 where the query phases depicted in Figure 2.2 are not allowed. Given a public key, the adversary simply outputs two equal length messages M_0, M_1 and the challenger responds with an encryption C^* of M_γ. The adversary wins if it can predict γ.

2.2 Identity-Based Encryption

The formal notion of an identity-based encryption scheme was developed in [155, 39]. An identity-based encryption scheme is specified by four probabilistic polynomial time (in the security parameter) algorithms: Set-Up, Key-Gen, Encrypt and Decrypt.

Set-Up: This algorithm takes as input a security parameter 1^κ, and returns the system parameters PP together with the master secret key msk. The system parameters include a description of the message space $\mathscr{M}$, the ciphertext space $\mathscr{C}$, the identity space $\mathscr{I}$ and the master public key. They are publicly known while the master secret key is known only to the private key generator (PKG). Usually, the descriptions of the different spaces are implicit in the description of the master public key and this itself is referred to as the public parameter PP.

Key-Gen: This algorithm takes as input an identity $\mathsf{id} \in \mathscr{I}$ together with the public parameters PP and the master secret key msk and returns a private key d_{id}, using the master key. The identity id is used as the public key while d_{id} is the corresponding private key.

Encrypt: This algorithm takes as input an identity $\mathsf{id} \in \mathscr{I}$, a message $M \in \mathscr{M}$ and the public parameters PP and produces as output a ciphertext $C \in \mathscr{C}$.

Decrypt: This takes as input a ciphertext $C \in \mathscr{C}$, an identity id, a corresponding private key d_{id} and the system parameters PP. It returns the message M or $\perp$ if the ciphertext cannot be decrypted.

These set of algorithms must satisfy the standard soundness requirement.

If
(PP,msk) is output by $\mathsf{Set\text{-}Up}$;
d_{id} is a private key returned by $\mathsf{Key\text{-}Gen}$ for an identity id;
C is a ciphertext produced by $\mathsf{Encrypt}$ on a message M,
using identity id and public parameters PP;
then
the output of $\mathsf{Decrypt}$ on C, id, d_{id} and PP should be M.

The comments regarding the encryption and decryption algorithms made in the context of PKE schemes are also applicable here. Additionally, similar comments apply to key generation. Given an identity and public parameters, it might be possible to have a set of corresponding decryption keys. In that case the key generation algorithm can be visualised as a strategy for sampling from this set. Note that the PKG can decrypt any message encrypted under any identity since it is the PKG who generated the private key for that identity. This is the so-called *key escrow* property of identity-based cryptography.

2.2.1 Hierarchical Identity-Based Encryption

Hierarchical identity-based encryption (HIBE) is an extension of IBE. The basic motivation for HIBE schemes is based on the following rationale. The generation of private key can be a computationally intensive task. The identity of an entity must be authenticated before issuing a private key and the private key needs to be transmitted securely to the concerned entity.

HIBE reduces the workload of the PKG by delegating the task of private key generation and hence authentication of identity and secure transmission of private key to its lower levels. However, only the PKG has a set of public parameters. The identities at different levels do not have any public parameters associated with them. Apart from being a standalone cryptographic primitive, HIBE has many interesting applications.

In contrast to IBE, for a HIBE identities are represented as vectors. So for a HIBE of maximum height h (which is denoted as h-HIBE) any identity id is a tuple

Chapter 4
Boneh-Franklin IBE and its Variants

The first practical identity-based encryption scheme using bilinear pairing is attributed to Boneh and Franklin [39]. They also came up with the security definition of IBE and a reductionist proof that their IBE scheme is secure in the proposed security model assuming the hardness of the Bilinear Diffie-Hellman problem (BDH). Sakai, Ohgishi and Kashahara [151] had independently proposed an identity-based non-interactive key exchange scheme using bilinear maps. The method of private key extraction in [151] is identical to the private key extraction in the IBE scheme in [39]. The work of Boneh and Franklin caught immediate attention of the crypto community and spurred further research in this area.

4.1 Boneh-Franklin IBE

Construction of the IBE scheme in [39] proceeds in two steps. In the first step a scheme called BasicIdent is developed and shown to be secure in the sense of IND-ID-CPA. The security analysis of this scheme showed how to simulate key extraction queries made by an adversary. In the next step, this was further developed to obtain a scheme, called FullIdent, which is secure in the sense of IND-ID-CCA. In both schemes, certain hash functions are used and the security reduction models these hash functions as random oracles.

We first describe BasicIdent with an intuitive explanation of its security. This is followed by a more formal argument in terms of a security reduction.

Set-Up: Let $e : G \times G \rightarrow \Gamma_T$ be a symmetric bilinear pairing and P be a generator of G. Pick a random $s \in \mathbb{Z}_p^*$ and set $P_{\mathsf{pub}} = sP$. Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow G^*$, $H_2 : G_T \rightarrow \{0,1\}^n$. The master secret is s and the public parameters are $\mathsf{PP} = \langle P, P_{\mathsf{pub}}, H_1, H_2 \rangle$.

Key-Gen: Given an identity $\mathsf{id} \in \{0,1\}^*$, compute $Q_{\mathsf{id}} = H_1(\mathsf{id})$ and set the private key to $d_{\mathsf{id}} = sQ_{\mathsf{id}}$.

S. Chatterjee and P. Sarkar, *Identity-Based Encryption*, DOI 10.1007/978-1-4419-9383-0_4,

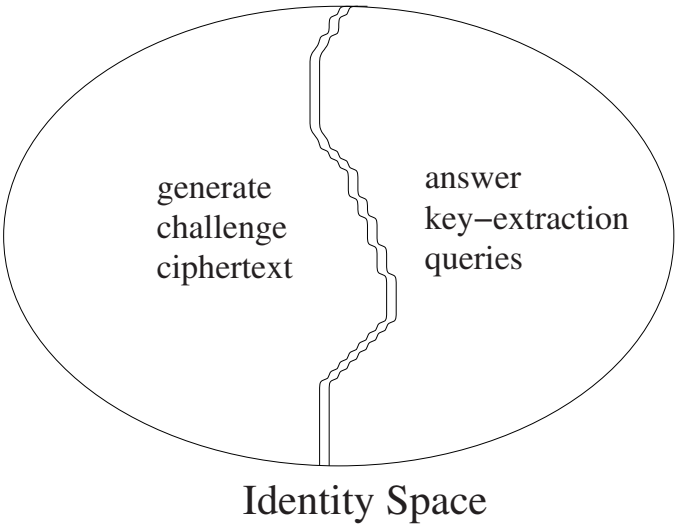

Fig. 4.1 Implicit partition of the identity space done by the security reduction.

Encrypt: To encrypt $M \in \{0,1\}^n$ to id compute $Q_{\mathsf{id}} = H_1(\mathsf{id})$, choose a random $r \in \mathbb{Z}_p^*$ and set the ciphertext:

$$C = \langle rP, M \oplus H_2(e(Q_{\mathsf{id}}, P_{\mathsf{pub}})^r) \rangle$$

Decrypt: To decrypt $C = \langle U, V \rangle$ using d_{id} compute

$$V \oplus H_2(e(d_{\mathsf{id}}, U)) = M$$

.

If C is an encryption of M under the public key id then we have

$$e(d_{\mathsf{id}}, U) = e(sQ_{\mathsf{id}}, rP) = e(Q_{\mathsf{id}}, sP)^r = e(Q_{\mathsf{id}}, P_{\mathsf{pub}})^r.$$

Hence the decryption algorithm returns M given a valid encryption under the public key of id.

Security analysis of BasicIdent shows that an adversary which can win the IND-ID-CPA security game for the scheme can be used to construct an algorithm to solve an instance of the BDH problem. At a high level the idea of the proof is the following. Given an instance of the BDH problem, the challenger sets up the IBE scheme and provides the public parameters of the PKG to the adversary. The solution to the BDH problem corresponds in some sense to the master secret key of the PKG. So, the challenger does not actually know the master secret key. But, the challenger has to answer key extraction queries made by the adversary and also generate a proper challenge ciphertext. The technical difficulty of the proof is in carrying out these two tasks. This is handled by randomly partitioning the identity space into two dis-